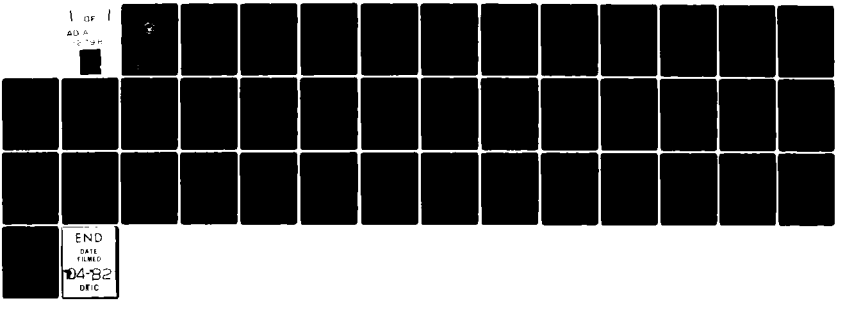
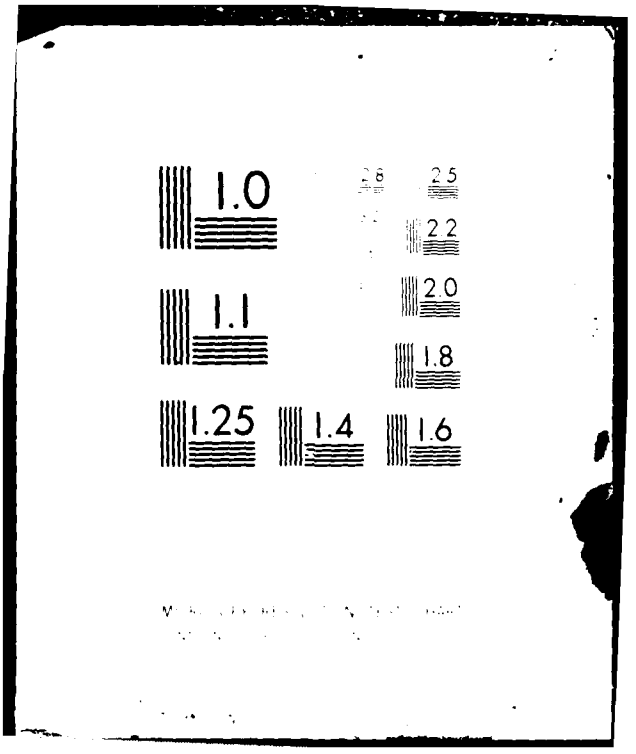
AD-A112 798 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2
A COMPUTER AIDED INSTRUCTION TUTORIAL FOR THE RAMTEK 9400 COLOR--ETC(U)
DEC 81 J R HAYES, J C SHERRARD

UNCLASSIFIED NL

1 OF 1
AD A
112798



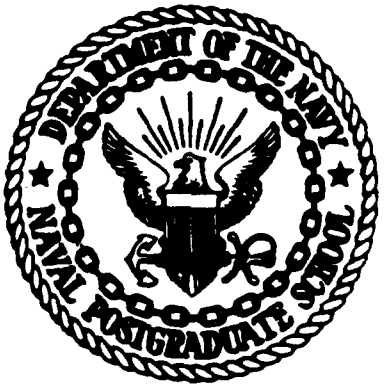
END
DATE FILMED
04-82
DTIC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5

(2)

AD A112 498

NAVAL POSTGRADUATE SCHOOL
Monterey, California



THESIS

A COMPUTER AIDED INSTRUCTION
TUTORIAL FOR THE
RAMTEK 9400 COLOR GRAPHICS DISPLAY SYSTEM
AT THE NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

by

John Robert Hayes

and

John Carr Sherrard

December 1981

Thesis Advisor: George A. Raye

Approved for public release, distribution unlimited.

DTIC FILE COPY

DTIC
ELECTE
APR 1 1982
S D
B

82 016

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER / 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) A Computer Aided Instruction Tutorial for the Ramtek 9400 Color Graphics Display System at the Naval Postgraduate School Monterey, California	5. TYPE OF REPORT & PERIOD COVERED Master's Thesis December 1981 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) John Robert Hayes John Carr Sherrard	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940	12. REPORT DATE December 1981 13. NUMBER OF PAGES 140
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report) 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Aided Instruction
CAI
Computer Graphics
Embedded Tutorial

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

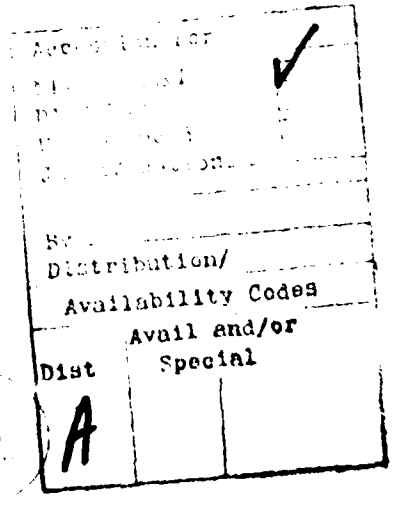
The objective of this thesis research was to create an embedded tutorial for the Ramtek-9400 Color Graphics Display System. An embedded tutorial is a self-paced Computer Aided Instruction (CAI) learning technique which is implemented on the actual equipment the subjects are learning to use. This technique has tremendous potential in helping the Department of Defense meet its goal of increased manpower effectiveness.

DD FORM 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

The Tutorial should eliminate the necessity for each user to possess and master large quantities of often confusing documentation. It should also free members of the laboratory staff from repetitive teaching tasks, allowing them to provide more specialized consulting. Goals of good CAI system design and man-machine interface, and the design techniques actually used to implement the embedded Tutorial are discussed. Evaluations by faculty, staff, and students were performed to identify areas of improvement. A User's Manual and Software Maintenance Manual were also written to accompany the embedded Tutorial.



Approved for public release, distribution unlimited.

A Computer Aided Instruction
Tutorial for the
Ramtek 9400 Color Graphics Display System
at the Naval Postgraduate School
Monterey, California

by

John Robert Hayes
Lieutenant Commander, United States Navy
B.S., Pennsylvania State University, 1971

and

John Carr Sherrard
Lieutenant Commander, United States Navy
B.S., Iowa State Univesity, 1971

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION SYSTEMS

from the

NAVAL POSTGRADUATE SCHOOL
December 1981

Author: _____

Author: _____

Approved by: _____
Thesis Advisor

_____
Second Reader

_____
Chairman, Department of Administrative Sciences

_____
Dean of Information and Policy Sciences

ABSTRACT

The objective of this thesis research was to create an embedded tutorial for the Ramtek 9400 Color Graphics Display System. An embedded tutorial is a self-paced Computer Aided Instruction (CAI) learning technique which is implemented on the actual equipment the subjects are learning to use. This technique has tremendous potential in helping the Department of Defense meet its goal of increased manpower effectiveness. The Tutorial should eliminate the necessity for each user to possess and master large quantities of often confusing documentation. It should also free members of the laboratory staff from repetitive teaching tasks, allowing them to provide more specialized consulting. Goals of good CAI system design and man-machine interface, and the design techniques actually used to implement the embedded Tutorial are discussed. Evaluations by faculty, staff, and students were performed to identify areas of improvement. A User's Manual and Software Maintenance Manual were also written to accompany the embedded Tutorial.

TABLE OF CONTENTS

I. INTRODUCTION ---------- 6

II. BACKGROUND ---------- 9

III. GOALS OF CAI ---------- 12

A. GENERAL FEATURES OF CAI PROGRAM DESIGN ----- 12

B. CONSIDERATIONS OF SOFTWARE ---------- 15

C. HUMAN COMMUNICATION NEEDS ---------- 17

IV. CURRICULUM DESIGN TECHNIQUES ---------- 20

A. DRILL AND PRACTICE ---------- 20

B. TUTORIAL ---------- 20

C. SIMULATION ---------- 21

D. PROBLEM SOLVING ---------- 21

E. INQUIRY/DIALOGUE ---------- 21

V. PROGRAM DESCRIPTION ---------- 23

VI. PROJECT DESIGN APPROACH ---------- 24

A. USERS NEEDS ---------- 24

B. TYPE OF INSTRUCTION ---------- 25

C. SOFTWARE DESIGN ---------- 26

D. DESIGN FEATURES ACHIEVED ---------- 28

VII. EVALUATION OF THE PROGRAM ---------- 31

VIII. CONCLUSIONS ---------- 33

IX. RECOMMENDATIONS ---------- 35

LIST OF REFERENCES ---------- 36

BIBLIOGRAPHY ---------- 38

INITIAL DISTRIBUTION LIST ---------- 40

I. INTRODUCTION

As more and more sophisticated computers with their specialized applications become available, the requirement for training the users to operate this equipment also increases. As the computer industry grows, so does the demand for technicians, programmers, and analysts. The industry has a difficult enough time filling those technical positions with qualified personnel; the supporting fields, such as training and documentation, often remain unfilled. This may be because of either the lack of qualified personnel or a management philosophy that does not see these support roles providing direct profits to the company. The result is that there are a large number of computers on the market with often poor or nonexistent documentation required for their use. Nowhere is this more evident than in computer graphics.

One of the highest priorities in the Department of Defense is the efficient use of manpower. One way in which manpower efficiency may be increased is to increase the productivity of personnel in the execution of their tasks. Another way is to raise the level at which seemingly unqualified personnel may be employed. This critical manpower requirement is met in part by developing methods whereby all operational computer-based systems teach the

potential operators how to interact with them. This research is an attempt to promote this critical field.

At the Naval Postgraduate School (NPS), Monterey California, the problem of incomplete documentation exists with a sophisticated color graphics system, the Ramtek 9400 Color Graphics Display System. The complexity and capability of the machine is extensive and could be of great value, not only to graduate student research, but also to fleet users. However, the documentation and instructions currently available are ambiguous, vague, and incomplete, resulting in underutilization of the equipment. The problem is to teach members of the NPS community the use of this highly sophisticated piece of equipment.

There are several possible ways of solving this problem. Structured courses could be taught several times a year for those interested students. This requires dedicated instuctors knowledgeable in the system, who can devote their time to this. Another possible solution is to rewrite the manuals, making them clearer and providing programming examples of how the software could work. Thus, students could teach themselves from a book, with additional guidance from the staff as necessary. Another approach is to develop an embedded tutorial which will provide the user with a self-paced method of developing the skills necessary to write graphics software while operating in the environment of the graphics computer. The Socratic or tutorial method

of intruction has long been acknowledged as an educational ideal. It is not normally used today because of its great cost. Computer augmented instruction regains the type of one-on-one relationship that is so desirable. [Ref. 1]

The use of Computer Aided Instruction (CAI) for developing application software is the subject of this thesis. The history and growth of CAI, its educational goals, and man-machine interface design are first reviewed. Next, different CAI curricular design techniques are examined, and the advantages and disadvantages of each outlined. The methods actually used in designing and implementing the Ramtek 9400 embedded Tutorial are then discussed, and finally a summary of actual evaluations of the developed Tutorial is presented.

II. BACKGROUND

There are many terms used with CAI. Before any discussion of the subject can occur, these different terms need to be defined. The following list is not inclusive, but it does cover the major terms used in the field today.

A. DEFINITIONS

CAI - Computer Aided Instruction- CAI is primarily used for direct instructional support. Typical examples of CAI are drill and practice, tutorials, simulation/gaming, inquiry/dialogue, information retrieval, and problem solving. [Ref. 2: p. 13] The computer is used as the means of problem solving. [Ref. 3: p. 10]

CMI - Computer Managed Instruction- CMI is primarily used for instructional support functions. Typical examples of CMI are testing, prescribing, recordkeeping, scheduling, monitoring, and time and resource management. [Ref. 2: p. 13]

CBI - Computer Based Instruction- The area of education that encompasses both CAI and CMI.

CAL - Computer Aided Learning- A broad application of computers in the education environment [Ref. 4: p. 11], similar to CBI.

Courseware - Computer software having educational content. [Ref. 4: p. 13]

User Station - User station is normally considered as the place where the student interacts with the computer, [Ref. 5: p. 17] usually a terminal. The user station can be geographically separated from the computer.

B. HISTORY

Throughout the past decade, more and more CAI systems have been used. To understand why, both the advance of computer technology and the teaching environment need to be discussed.

Until the advent of time-sharing systems, where a user could react interactively with a computer, usually through a terminal, there was no great incentive for CAI. Computers were run in the batch mode usually doing transaction type processing. As time-sharing systems came on line, CAI was developed. One of the first CAI systems was the PLATO sytem at the University of Illinois. [Ref. 6: p. 15] The Navy also started using CAI during the same time period as PLATO for instructing large numbers of recruits in technical training classes. [Ref. 7] These systems required the use of a large expensive host computer which virtually eliminated all but government organizations and research projects from using CAI. As technology advanced, the increased use of lower cost mini-computers allowed more and more of the larger school systems and universities to consider using CAI. [Ref. 8: p. 73] The response time of the interactive terminals were acceptable, and a number of

terminals could be connected to each mini-computer. However, the smaller schools could not justify the cost of these systems. With micro-computers, the cost of providing several terminals for interactive use became acceptable. [Ref. 9: p. 105] Thus, many CAI systems are now being used in schools throughout the country.

Another reason for the development of CAI is the shortage of qualified teachers. With the increased school population of the late 60's and early 70's, alternatives for providing instruction were necessary. Colleges were teaching large introductory courses in auditoriums and individualized instruction was lacking. The cost of hiring the additional staff necessary to reduce class size was prohibitive, so schools were being forced to look for alternative ways of providing the individualized instruction. [Ref. 10: p. 289] During this period, the Navy had similiar problems with providing qualified instructors to teach recruits fundamentals while at the same time transitioning to an all volunteer force. Computer aided instruction was a primary alternative. [Ref. 11: p. 16]

III. GOALS OF CAI

Throughout the literature concerning CAI, man-machine interface, and software design, there are many design considerations which reappear. The goals of designing good CAI programs should consider all of them. This section will discuss many of the goals as they apply to their respective disciplines.

A. GENERAL FEATURES OF CAI PROGRAM DESIGN

One of the main advantages of CAI is its ability to adapt instruction to the individual needs of the learner. To achieve this goal Caldwell recommends: [Ref.12: pp. 7-8]

1. Learner Control

Learner control over the instructional sequence should be available. Except for instances where sequential instruction is necessary, the learner should have the ability to control the sequence of instruction. He should be able to back-up in the program and review previous frames, if desired. He should be allowed to skip portions with which he is already familiar, and go directly to those areas where he needs additional instruction. Students should be given the opportunity to advance, review, and exit lessons, except where such control defeats the pupose of the lesson. The ability of learners to pace themselves provides

a degree of individualization not present in purely linear (sequential) programs.

2. Learning Environments

Adaptive and responsive learning environments should be provided. By allowing self-pacing and individualized branching, learners are helped to select the pathway through the materials that is most appropriate for their needs.

3. Modularized Segments

Modularized segments, arranged in hierarchical patterns allow for great flexibility in program implementation. Each lesson can be built and implemented without the need for other sections of the program. Curriculum materials can specifically address each necessary skill in a defined context area. This can be done in a manner which provides for developement of skills not mastered. It also allows the bypassing of skills which have already been mastered, which in-turn can reduce student frustration and increase curriculum effectiveness.

4. Performance Objectives

All skills to be mastered should be stated in performance objectives. The measure of success of a program should be based upon these performance objectives. They allow for the precise diagnosis of skills already mastered, identification of skill deficiencies, and an evaluation of learner progress.

5. User Progress

Progress should be measured in terms of the mastery of performance objectives.

6. Diagnosis and Prescription

Strategies for diagnosis and prescription should be used. As each learner is diagnosed in respect to his skill level, he can be placed appropriately within the curriculum, with the best materials needed to further his progress.

7. Multisensory Programs

Multisensory formatted programs should be used whenever possible.

Impementation of the above guidelines require more specifics. One of the poorest ways of motivating a student is to require him to read line after line of text on a video terminal. Many techniques are available to ease this monotony. One of these techniques is the ability to double-space lines of text. This allows for enhanced visual effect. A color graphics system such as the Ramtek 9400 has a multitude of techniques not available on alphanumeric terminals. The use of color to set off sections of the screen can cause the learner to concentrate on that section, or can cause other sections to be ignored. Reverse highlighting is a good technique. When areas of text are no longer being reviewed by the user, then they should be erased. This will keep the screen from becoming overcrowded and prevent slow readers from being discouraged.

Animation, graphic displays, or anything to create variety and interest in the display will help contribute to effective instructional programs. The user should be shown instead of told. The use of graphics and the ability to erase, rewrite, and blink parts of the screen, can make concepts much clearer than long displays of text. Graphics can be used to dynamically illustrate abstract concepts or invisible processes. [Ref. 11: p. 17] By providing interactive lessons, the user will become involved in the program. Simulation and dialogue programs are the best instructional strategies for promoting interaction. [Ref. 12: p. 8] Menus, performance options, and prompts also provide for interaction. Task description is necessary to prevent users from becomming confused. Users should be told what to expect throughout the course of the program. Better yet, examples or simple exercises can lessen or even eliminate task confusion. [Ref. 12: pp. 9-10]

B. CONSIDERATIONS OF SOFTWARE

Designing software for graphical applications within CAI requires many of the features described above. However, due to the additional complexity of displaying graphics as well as text, special considerations must be given.

1. Interactiveness

Interactiveness must be considered when complex graphical displays are required. Many line drawings, color filling and other graphical displays take time. This time

lag between a user's input and the graphical response could reduce the interactiveness of the system. The areas where this is especially true are in the older systems where long computational times were required. There is a well known principle that learning requires activity on the part of the learner, and if a learner is spending most of his time staring at a blank screen, the interactiveness of the system is deficient.

2. Responsiveness

Responsiveness works closely with interactiveness. If the time for a computer to respond to a user's input is excessive, then he will tend to lose interest. If a program requires long computational time, periodic messages telling the user that he is not forgotten are encouraged. Acceptable response times vary as to users and how they perceive it. [Ref. 13: p. 76] Studies by R. B. Miller suggest that response times for complex interaction should be less than a minute. [Ref. 14: p. 267] Others say that users interest while operating on an interactive terminal, will wane if response times for normal actions take more than 3-15 seconds. [Ref. 15: p. 134]

3. Error Recovery

Error recovery is another challenge for the software designer. Whenever unfamiliar users are involved in interactive programs, there exists the chance for them to hit an improper key in the middle of a session. If the

computer response to that is terminating the execution of the program due to a fatal error, student frustration increases immensely. Fail-soft provisions to handle such situations should be incorporated with friendly reminder messages telling the user that his input was incorrect. "Humble" phrasing of error messages is important if the user's ill-will is not to be incurred. [Ref. 16: p. 177]

4. Program Access

Program access should be made easy for the user. If the user must proceed through a long bureaucratic bookkeeping process before being able to run the desired program, he will become frustrated and loose interest in the process. Ideally, the student should be able to turn on the terminal, enter an account number, password, name of a desired program and be able to start executing the program within seconds. [Ref. 10: p. 293]

C. HUMAN COMMUNICATION NEEDS

One of the main problems with man-computer communications is that computer systems typically respond only to inputs phrased exactly as required by an artificial language. When used incorrectly, the best response might be an error message stating the problem. When people communicate with each other, other forms of communication such as body language, eye contact, and expressions provide much of the meaning. The graceful system should be able to provide some of the same capabilities. Hayes, Ball and

Reddy describe some of the aspects that a graceful system should have. [Ref. 17: p. 22]

1. Flexible Parsing

The small mistakes people usually make when using any language, whether natural or artificial, should not prevent the receiver from interpreting the meanings. The graceful system should be able to handle minor spelling and syntactical errors. If the system cannot accomodate them, then the user should be shown what the computer thinks he means, and allow him to select the proper meaning.

2. Robust Communication

The program should be able to tell the user when it does not understand him and more importantly, be able to tell when the user does not understand the program. This should be done without execcive verbosity or unnecessary disruption of the flow of conversation. It should let the user know what assumptions the program is making concerning the user's interpretations.

3. Focus Tracking

The program should be able to keep track of the flow of the program, and allow for the user to branch from one area to several different ones, and then back to the original. At the original area, the program should be able to return to the same context achieved before branching.

4. Natural Output

A graceful system should provide the output in the same context as the user is working. If the proper output is a short precise answer, then that is what should be provided. If the user expects a long detailed description of something, then the program should be able to provide it.

5. Explanation Facility

The system should be able to explain both statically and dynamically what the program can, and is doing. Static explanation could be in the form of a help routine which explains what the program can and can not do. A dynamic facility would explain to the user what the program is doing at that time, why, and how it is being done.

6. Personalization

Each user is different and a graceful system should be able to adjust to each. It should be able to recognize and correct errors that a user typically makes, and if desired, let the user know what he is doing wrong. Be using his name, the user will not feel that he is talking to an impersonal machine.

The above desired features of a system are hard to achieve unless the program is designed with each in mind. If they all can not be implemented, each should be considered in trying to create a more graceful system.

IV. CURRICULUM DESIGN TECHNIQUES

There are many different ways of designing systems to instruct students. Different considerations are needed for each method. Most entail defining the target population and the learning needs of the students. Additonally, the courseware, hardware, and software available/needed must be defined. Finally, the decision of implementing the Computer Aided Instruction (CAI) program along with the more administrative aspects of Computer Managed Instruction (CMI) must be considered. There are several different types of instruction.

A. DRILL AND PRACTICE

This type of instruction is designed to reinforce previously taught material. No new information is normally presented. Exercises are given to the students and may be in a fixed format, randomly generated, or tailor-made for each student.

B. TUTORIAL

A tutorial is normally used to introduce new information. Independent lessons are normally used which will allow a user to skip lessons previously covered, or to review material which is unclear.

C. SIMULATION

A simulation program allows manipulation of a model of a real-life (or theoretical) situation or process. It may relieve the student of tedious numerical work, expand or compress time, or permit the study of systems or experiments too expensive, massive, delicate, dangerous or otherwise unavailable for practical utilization. Two examples of instructing by computer simulation are aircraft landing systems and nuclear power plant emergency procedures.

D. PROBLEM SOLVING

Whether the intent is to study the computer itself or to use it as a device in work or study, the student will program and otherwise control the system. It then may be utilized to manipulate data input by the student, or in some cases, to control or gather its own data from an experiment. In doing so, students learn to use the computer as a tool in the discipline they are studying.

E. INQUIRY/DIALOGUE

Inquiry is an interaction between student and computer, where both student and computer alternately asks questions and requires answers of the other. The dialogue may be simply structured, with a highly documented logic train, or it may be a more sophisticated program which facilitates inquiry by the student in a natural language. [Ref. 13: pp. 3-4]

These different techniques can be combined and used throughout a program. A tutorial can first be used to present the information to the student. A drill and practice session could then be used to reinforce the concepts. A simulation of the model described previously could be run, thus showing the user what was described. These are merely different ways of providing CAI to the user.

V. PROGRAM DESCRIPTION

The principle objective of the proposed research is to create an embedded tutorial for the Ramtek 9400 Color Graphics Display System. The Tutorial is to be an effective aid to members of the Naval Postgraduate School faculty, staff and student-body in the use of the equipment and related color graphics techniques. It should eliminate the necessity for each user to possess and master large quantities of documentation. By providing both introductory and refresher courses in the use of the equipment, it will free members of the Laboratory staff from repetitive teaching tasks, allowing them to provide consulting on more specialized topics. Such a combination of computer-based tutorials and individualized instruction has been found to be highly successful. [Ref. 19]

VI. PROJECT DESIGN APPROACH

When designing the system of instruction for the Ramtek 9400, the theoretical considerations were evaluated. Those which were most applicable were included while others were reserved for further research. Only CAI was to be designed, as the number of students which would realistically use the Ramtek did not warrant a CMI approach. The items which were considered are listed below.

A. USERS NEEDS

The first step in designing a CAI program was to consider the users of the instruction. Their ability level and programming background were evaluated so as to best design the program. A constraint that had to first be considered was the requirement to work with the FORTRAN language, since the Ramtek interface was in FORTRAN. The Naval Postgraduate School had a FORTRAN-based structured programming course taught to most students entering the school. Therefore, the program of instruction was to be directed towards students having a basic knowledge of FORTRAN. The use of the DEC PDP 11/50 with the RSX-11M Operating System would normally have been unfamiliar to most students, as the basic course was taught on an IBM system. Therefore, more detail of the RSX-11M operations would be necessary, with just the aspects of FORTRAN peculiar to RSX-

11M emphasized. There are more than two hundred different FORTRAN commands which are used to operate the Ramtek 9400. A program designed to teach students just the basic instuctions would have no need for all of them; therefore, the more complex and seldom used commands were not considered applicable.

B. TYPE OF INSTRUCTION

The different types of curricular design techniques listed earlier were reviewed as to their applicability in teaching students how to use and program the Ramtek 9400. The tutorial method seemed the most logical for this type of instruction and was the method chosen. Several different ways of presenting the material were possible, and after several iterations of program content, two major areas of interest were defined. One area covered all of the items and commands necessary to familiarize the student with operating the Ramtek 9400. This area included review of the RSX-11M Operating System logon procedures, file manipulation, and the steps necessary to write, compile, and execute a program. It also described the Ramtek hardware.

The second half of the tutorial consisted of the different Ramtek FORTRAN instuctions. These instructions were grouped into color, graphics, text writing, cursor manipulation, and image categories. A Users Manual was also written to provide written backup for each of these areas. [Ref. 20]

C. SOFTWARE DESIGN

Designing a tutorial incorporating the concepts mentioned above, such as user control, focus tracking, personalization, pictures instead of text, etc. required developing an algorithm which would act as a "control" shell to call the different instructional modules, but would be transparent to the user. The different options which were considered necessary to provide the user with control over the instructional sequence of the tutorial are listed below.

1. Sequential

The tutorial was to allow the user to be able to step through the entire program in sequence if desired.

2. Scroll Back

If, after completion of a certain subject, the user wished to back up and review it, then a scroll-back option was available.

3. Direct Branch

If the user wanted to branch to any specific module, then by typing the name of the desired module, he should be able to go directly to that module.

4. Help

A help menu would list the different subject areas available plus the proper commands necessary to get to them. The help menu was required to be accessable from any point within the tutorial. After calling the help module, the program should return the user to the module from which it

was called. If within the help module, the user desired to branch to another module, then that option was also possible.

5. Quit

The last option was the quit option. The user could choose to quit anywhere within the program. The tutorial would prompt the user to make sure his desire was to quit. This prevented inadvertent program termination.

One of the main problems of designing an algorithm to handle these techniques was the problem of incorporating both graphical and textual material. Several pages of text could appear with one graphical presentation, or different graphical presentations could appear with the same text displayed. To solve the conflicts of scrolling back into the end of the previous module, where the text might not match the graphical data presented, the decision was made to always enter a module at its beginning. By keeping each module small, only a few pages of text would need to be presented before arriving at the desired page. Scrolling back within a module would not be a problem for the outer "shell" program.

The resulting algorithm used to implement the main "shell" is based on a hierarchial tree structure, with forward and backward pointers connecting each different module. The algorithm polls the keyboard to determine the user's desires. It then calls one of five subroutines. The

first subroutine allows the user to step through the program sequentially. The second allows the user to scroll back from the current location to the previous page. If the Tutorial is at the beginning of a module, it then follows a pointer to the previous module, setting a branch flag. The third subroutine allows the user to select a new module name. It converts the name to a number and sets a branching flag, which then allows the Tutorial to branch directly to that module. The direct branch routine accepts the first four characters of the module name typed in, and uses an inverted file technique to determine the location of the module within the program. A branch flag is set and passed to the calling routine to allow the main program to act appropriately. The other two options are to call a help menu and to call for a quit routine.

The shell was tested extensively to ensure all of the logic operated as desired before writing any instructional modules. Future modules can easily be added anywhere within the instructional program by changing the appropriate pointers and making several other minor adjustments. A Software Maintenance Manual was written to assist in adding additional instructional lessons to the Tutorial or making any other required alterations to the program logic.

D. DESIGN FEATURES ACHIEVED

The design features which were achieved by this program are listed below.

1. CAI Techniques

Since the Tutorial was designed to use CAI and not CMI techniques, no administrative aspects were included in the program.

2. User Control

Due to a design consideration, modules can only be entered from the beginning. Thus, the only part of the program when a user does not have complete control is during the process of scrolling back from one module to the previous one.

3. Interactiveness

The program requires constant user inputs, thus allowing the user to become involved, thereby increasing the interactiveness of the program.

4. Responsiveness

The response time of the Ramtek 9400 is well within the 2-10 second range unless the host PDP 11/50 is overloaded with other CPU intensive programs.

5. Error Recovery

The error recovery features built into the shell allow the user freedom of input without "crashing" the system. The worst-case situation is when the user is returned to the place of improper input with a message stating the problem and acceptable answers.

6. Accessability

Access to the Tutorial requires the user to log on to the system, enter a password, access the proper directory, and run the program. Detailed instructions are provided in the users manual for that purpose.

Techniques which were not used included the more advanced ideas for diagnosing user skill levels and for providing outputs appropriate to each user. These were considered to be beyond the scope of the project.

VII. EVALUATION OF THE PROGRAM

The completed Tutorial was evaluated by randomly-selected students and faculty members. The experience level of the evaluators ranged from students and faculty expert with the Ramtek 9400 to persons with just basic knowlege of FORTRAN, but no knowledge of either the RSX-11M Operating System or the Ramtek 9400. At the completion of their first session with the Tutorial, comment sheets were provided to record their reaction. Their responses are summarized below.

Evaluators who were not experienced users were impressed by the amount of examples and illustrations that were provided in the Tutorial. They felt that they knew where they were within the Tutorial most of the time. The tree diagram in the HELP menu, along with individual page identification, were mentioned as being the primary reasons for this. The explanation of the material covered was considered good, but they indicated that the Tutorial provided too much text for them to read. The students were divided as to whether the mechanics of using the Tutorial (four function keys and the carriage "return") were confusing. However, they all thought the directions provided were clear and concise. The students thought the Tutorial should be expanded, but were hesitant to state that

they could immediatly write a simple graphics program for the Ramtek 9400. They all thought it was enjoyable, and that it would be of great benefit to future Ramtek users. Most inexperienced users felt "overpowered" by the sheer quantity of information presented in the Tutorial.

The more experienced users generally agreed with the comments mentioned above. They realized that a great deal of text was necessary to fully and accurately explain the the Ramtek software. As expected, they seemed to have less problems with the mechanics of using the Tutorial. They all saw the benefits of having a self-paced tutorial available for student learning and realized the potential for having a "ready-reference" program where they could easily reacquaint themselves with specific aspects of the system.

The average time for a beginning student to complete the Tutorial was approximatly two hours. This first-time session primarily gave the students an overall familiarity with the Ramtek 9400. The more experience users completed the session in about one hour, as they tended to skim over familiar areas. Surveys of subsequent sessions were not given. These sessions are expected to deal with more specific areas within the Tutorial.

VIII. CONCLUSIONS

The desired goal of providing an embedded Tutorial for teaching beginning students graphical programming techniques was achieved. A Users Manual and a Software Maintenance Manual were also written to accompany the embedded tutorial. The thesis research, interalia, indicates that CAI can be used in a highly productive way to teach computer graphics, and, conversely, computer graphics offers unique capabilities to increase the effectivenes of CAI.

More advanced CAI concepts, such as simulation, problem solving, and inquiry/dialogue, were not implemented nor contemplated. The embedded Tutorial appeared to be the best method of utilizing CAI to achieve the stated objectives. A drill/practice type instruction format could easily be added at the end of each module to reinforce the concepts presented.

The software was designed to provide expandability and maintainability. The ease of expanding the Tutorial will allow for modules to be added or deleted as the need arises.

Judging by the initial favorable evaluations conducted by first-time users, the embedded Tutorial will become a useful addition to computer graphics education at the Naval Postgraduate School and will result in cost and manpower savings in teaching students how to use the Ramtek 9400. As

procurement of this graphics device is contemplated for fleet users, the Tutorial may have additional applications Navy-wide. With added instructional lessons, it will provide increased cost savings and manpower utilization.

IX. RECOMMENDATIONS

Additional modules should be added to the Tutorial. As more students use and understand the Ramtek 9400, more of the advanced concepts can and should be presented.

Additional Computer Managed Instruction (CMI) modules should be added to the beginning and end of the program. They can keep track of each user, determine which modules are called during each sequence, and provide the computer staff with information to determine which areas of the Tutorial will need to be improved.

There should be a comment file set up allowing users to record their thoughts and ideas as they occur. Periodic review of the comment file by the programming staff is necessary to ensure the comments are acted upon.

LIST OF REFERENCES

1. Suppes, Patrick, "The Uses of Computers in Education", Scientific American, v.215, no. 3, pp. 206-220, September, 1966.

2. Milner, Stuart D., "How to Make the Right Decisions About Microcomputers", Instructional Innovator, v. 25, no. 6, pp. 12-19, September 1980.

3. Muscat, Eugene, "Microcomputers in Business Education", Business Education World, v. 60, no. 4, pp. 10-11, March-April 1980.

4. Aiken, Robert M., Braun, Ludwig, "Into the 80's with Microcomputer-Based Learning",Computer, v.13, no.7, pp. 11-16, July 1980.

5. Bork, Alfred, "Machines for Computer-Assisted Learning", Educational Technology, v. 18, no. 4, pp. 17-20, April 1978.

6. Magidson, Errol M.,"Student Assessment of PLATO: What Students Like and Dislike About CAI", Educational Technology, v. 18, no. 8. pp. 15-19, August 1978.

7. Mayo, G. Douglas, "Computer Based Instructional Systems", Journal of Educational Technology Systems, v. 2, no. 3, pp. 191-201, Winter 1974.

8. Su, Stanley Y.W., Eman, Ahmed E., "Teaching the Principles of Software Systems on a Microcomputer", Journal of Computer-Based Instruction, v. 1, no. 3, pp. 73-79, February 1975.

9. Attala, Emile,"CAI Debut in the World of General Software", Journal of Computer-Based Instruction, v. 3, no. 4, pp. 103-106, May 1977.

10. Gerhardt, Lester A., Carlson, A.B., "Instructional Computer Graphics in Electrical and Systems Engineering",Frontiers in Education, Proceedings of the 8th Annual Conference, pp.288-295, 1978.

11. Crawford, Alice M., "Interactive Computer Graphics for Simulation Displays", Proceedings of the Human Factors Society, 21st Annual Meeting, 1977.

12. Caldwell, Robert M., "Guidelines for Developing Basic Skills Instructional Materials for Use with Microcomputer Technology", Educational Technology, v. 20, no. 10, pp. 7-12, October 1980.

13. Bevan, Nigel, "Is There an Optimum Speed for Presenting Text on a VDU", International Journal of Man-Machine Studies, v. 14, pp. 59-76, 1981.

14. Miller, R.B., "Response Time in Man-Computer Transactions", AFIPS Conference Proceedings, 1968,FJCC, v. 33, pp.267-277, 1968.

15. Carlson, Eric D.,"Graphics Terminal Requirements for the 1970's", Computer, v. 9, no. 8, August 1976.

16. Cuff, Rodney N., "On Casual Users", International Journal of Man-Machine Studies, v. 12, pp. 163-187, 1980.

17. Hayes, Phil, Ball, Eugene, Reddy, Raj, "Breaking the Man-Machine Communication Barrier", Computer, v. 14, no. 3, pp. 19-30, March 1981.

18. Wright, Robert B. Jr., "A Comparison Between Various Levels of Instructional Design and the Interactive Computer System Required to Deliver or Support Each Level", Association for Educational Data Systems, Proceeding of the 15th Annual Convention, pp.1-6, 1978.

19. Chambers, J.A., Sprecher, J.W., "Computer Assisted Instruction: Current Trends and Critical Issues", Communications of the ACM, v. 23, p.332, 1980.

20. Sherrard, John C., Hayes, John R., "Users Manual for the Ramtek 9400", Naval Postgraduate School, Monterey, Ca., December 1981.

BIBLIOGRAPHY

Atkinson, Richard C. and Wilson, H. A., Computer-Assisted Instruction: A Book of Readings, Academic Press, 1969.

Bork, Alfred, "Machines for Computer-Assisted Learning", Educational Technology, v. 18, no. 4, pp. 17-20, April 1978.

Gable, Alice, Page, Carl V., "The Use of Artificial Intelligence Techniques in Computer-Assisted Instruction: An Overview", International Journal of Man-Machine Studies, v. 12, pp. 259-282, 1980.

Gaines, Brian R., "The Technology of Interaction-Dialogue Programming Rules",International Journal of Man-Machine Studies, v. 14, pp.133-150, 1981.

Gille, Frank H., Holdaway, Douglas E., and Waldman, Harry, Automated Education Handbook, Automated Education Center, 1977.

Henry, Robert D., Howard, James A., "SAL- A Simple Author Language For a Microcomputer Assisted Instruction System", Journal of Computer-Based Instruction, v. 3, no. 4, pp. 107-114, May 1977.

Hickey, Albert E., Computer-Assisted Instruction: A Survey of the Literature, ENTELEK Incorporated, 1968.

Lee, John D., Lee, Timothy D.,"Learning Geography With a Computer", Personal Computer World, v. 1, no. 7, pp. 37-38, November 1978.

Lower, Stephan K.,"Maxi Authoring Languages in the Era of the Mini",Journal of Computer-Based Instruction, v. 3, no. 4, pp.115-122, May 1977.

Lukey,F.J.,"Understanding and Debugging Programs", International Journal of Man-Machine Studies, v. 12, pp. 189-202, 1980.

Mayo, G. Douglas, "Ten Pitfalls in CMI Development", Journal of Computer-Based Instruction, v. 1, no. 3, pp. 70-72, February 1975.

Moss, Ronnie L., "I/O-Computer Scientist and Academia", A.E.D.S. Monitor, v. 12, no. 10, pp.15-24, May-June 1974.

Myers, Ware, "Computer Graphics: Reaching the User", Computer, v. 14, no. 3, pp. 7-18, March 1981.

Nievergelt, J., "XS-0: A Self-Explanatory School Computer", A.E.D.S. Monitor, v. 16, no. 10-12, pp. 10-13, April-June 1978.

Oldknow, Adrian, "Some Notes from a Microcomputer Workshop", Mathematics Teaching (GB), no. 91, pp. 45-47, June 1980.

Pressman, Rodger S., "Generalized Engineering Instuction using Interactive Computer Graphics", Frontiers in Education, Proceeding of the 8th Annual Conference, pp. 296-305, 1978.

Ramtek Corporation Publication No. 504607, Revision B, RM-9400 Series Graphic Display System Software Reference Manual, 1979.

Ramtek Corporation Publication No. 504637, Revision B, RM-9400 FORTRAN Interface Package (FIP), Software Reference Manual, 1980.

Schwab, Wallace, St.-Denis, Richard, "A Data Bank Experience on the Plato System", Journal of Computer-Based Instruction, v. 7, no. 2, pp. 41-46, November 1980.

Seidel, Robert J., and others, "Learner Control of Instructional Sequencing Within an Adaptive Tutorial CAI Environment", Instructional Science, v. 7, no. 1, pp. 37-80, January 1978.

Shneiderman, B., "Measuring Computer Program Quality and Comprehension", International Journal of Man-Machine Studies , v. 9, no. 4, pp. 465-478, February 1975.

Tennyson, Robert D., Rothen, Wolfgang, "Management of Computer-Based Instruction: Design of an Adaptive Control Strategy", Journal of Computer-Based Instruction, v. 5, no. 3, pp. 63-71, February 1979.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 54 Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
4.	Director of Computer Laboratories, Code 052Ra Department of Computer Science Naval Postgraduate School Monterey, California 93940	4
5.	LCDR John Robert Hayes, USN, Code 54Ht Department of Administrative Sciences Naval Postgraduate School Monterey, California 93940	1
6.	LCDR John Carr Sherrard, USN 5845 525th Avenue, West Oak Harbor, Washington 92377	1

END

DATE
FILMED

04-82

DTIC